

AD A 042003

Naval Warfare Research Center ✓

Technical Note  
NWRC-TN-37

April 1972

## SIMULATION MODELS OF SEARCH IN THE PRESENCE OF DECOYS

By: E. L. WONG

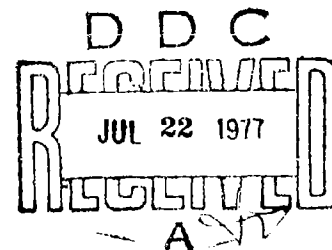
Prepared for:

NAVAL ANALYSIS PROGRAMS (Code 462)  
OFFICE OF NAVAL RESEARCH  
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-71-C-0119  
Task No. NR 274-008-34

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Approved for public release; distribution unlimited.



STANFORD RESEARCH INSTITUTE  
Menlo Park, California 94025 • U.S.A.

DDC FILE COPY

### TECHNICAL REPORT

A Technical Report is a document of Stanford Research Institute that presents results of work directed toward specific research objectives. The report is a comprehensive treatment of the objectives, scope, methodology, data, analyses, and conclusions, and presents the background, practical significance, and technical information required for a complete and full understanding of the research activity. Technical Reports are reviewed and approved by a division executive director or higher official of the Institute.

### RESEARCH MEMORANDUM

A Research Memorandum is a document of Stanford Research Institute that presents the results of work in progress. The purpose of the Research Memorandum is to invite comment on research in progress. It is a comprehensive treatment of a single research area or of a facet of a research area within a larger field of study. The Memorandum presents the background, objectives, scope, summary, and conclusions, as well as method and approach, in a condensed form. The report is reviewed and approved by a department director or higher official of the Institute.

### TECHNICAL NOTE

A Technical Note is a working paper that presents the results of research related to a single phase or factor of a research problem. The purpose of the Technical Note is to instigate discussion and criticism. It presents the concepts, findings, and/or conclusions of the author. The report is reviewed by a project leader or higher official of the Institute.



STANFORD RESEARCH INSTITUTE  
Menlo Park, California 94025 U.S.A.

Naval Warfare Research Center

April 1972

Technical Note  
NWRC-TN-37

## SIMULATION MODELS OF SEARCH IN THE PRESENCE OF DECOYS

By: E. L. WONG

Prepared for:

NAVAL ANALYSIS PROGRAMS (Code 462)  
OFFICE OF NAVAL RESEARCH  
ARLINGTON, VIRGINIA 22217

CONTRACT N00014-71-C-0119  
Task No. NR 274-008-34

SRI Project 1016-245

Approved for public release; distribution unlimited.

ACCESS ON TOP	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Blue Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
RE	
DISTRIBUTION AVAILABILITY CODES	
Dist.	AVAIL. CODE or SPECIAL
A	

UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified.

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute Menlo Park, California 94025		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP ---	
3. REPORT TITLE Simulation Models of Search in the Presence of Decoys			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Note,			
5. AUTHOR(S) (Last name, Middle initial, Last name) Ellen L. Wong			
6. REPORT DATE April 1972		7a. TOTAL NO. OF PAGES 74	7b. NO. OF REFS 1
8. CONTRACT OR GRANT NO. N00014-71-C-0119 RF 018-02-06 NR 274-008-34		9a. ORIGINATOR'S REPORT NUMBER(S) NWRC-TN-37	
		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited			
11. SUPPLEMENTARY NOTES (16) F01-V-1		12. SPONSORING MILITARY ACTIVITY Naval Analysis Programs (Code 462) Office of Naval Research Arlington, Virginia 22217	
13. ABSTRACT A simulation model that represents a submarine's search for a high value target within a specified operation has been developed. The model, constructed as an adjunct to the formulation and implementation of a computationally more efficient analytical model, supported the assessment of the potentials of tactical deception techniques in anti-submarine warfare. This technical note describes the details of the model structure. The assessment results are published in a separate, classified, final project report.			

DD FORM 1473 (PAGE 1)

S/N 0101-807-6601

UNCLASSIFIED  
Security Classification

253 000

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Modeling						
Simulation						
Submarine search process						
Antisubmarine warfare						
Decoys						
Submarine motion						
Semi-Markov models of search						
Submarine search for a HV						

## PREFACE

---

The work reported in this technical note was conducted as a sub-task within a larger project directed toward the assessment of tactical deception in antisubmarine warfare. The project was sponsored by Naval Analysis Programs, Mr. R. J. Miller, Director, in the Office of Naval Research. Mr. J. G. Smith was the ONR Project Scientific Officer.

The research effort was performed by the Naval Warfare Research Center, Mr. L. J. Low, Director, of Stanford Research Institute. Mr. A. Bien of NWRC was the project leader.

The author wishes to thank Mr. M. W. Zumwalt for his invaluable help in the preparation of the simulation model description.

# CONTENTS

---

DD Form 1473 . . . . .	iii
PREFACE . . . . .	v
LIST OF ILLUSTRATIONS . . . . .	vii
LIST OF TABLES . . . . .	viii
1. INTRODUCTION . . . . .	1
2. GENERAL SIMULATION MODEL CHARACTERISTICS . . . . .	2
2.1 Operational Framework . . . . .	3
2.2 Model Time . . . . .	4
2.3 Submarine/HVT/Decoy Motion . . . . .	5
2.4 Detection of the HVT and/or Decoy . . . . .	8
2.5 Input Parameters . . . . .	9
2.6 Submodel Outputs . . . . .	11
3. CONTINUOUS SEARCH - NO DECOY SUBMODEL . . . . .	13
4. SPRINT/DRIFT SEARCH - NO DECOY SUBMODEL . . . . .	16
5. CONTINUOUS SEARCH - DECOY FIELD SUBMODEL . . . . .	21
APPENDIX A Program Listing for Continuous Search - No Decoy Submodel (Mod 1) . . . . .	A-1
APPENDIX B Program Listing for SPRINT/Drift Search - No Decoy Submodel (Sprint) . . . . .	B-1
APPENDIX C Program Listing for Continuous Search - Decoy Field Submodel (Mod 3) . . . . .	C-1
DISTRIBUTION LIST	

## ILLUSTRATIONS

---

Fig. 2-1	Typical Search Initiation Situation . . . . .	6
Fig. 3-1	Sample Plot of HVT and Submarine Motion . . . . .	14
Fig. 3-2	Sample Results From the Continuous Search - No Decoy Submodel . . . . .	15
Fig. 4-1	Sample Plot of Motions of HVT and Submarine in the Sprint/Drift Model . . . . .	17
Fig. 4-2	Sample Results From the Sprint/Drift Search - No Decoy Submodel . . . . .	20
Fig. 5-1	Sample Plot of Motions of Four Decoys ( $D_1, \dots, D_4$ ), Submarine, and HVT From a Test Run Where All Except the Submarine are Prepositioned Initially . . . . .	24
Fig. 5-2	Sample Results From the Continuous Search - Decoy Field Submodel . . . . .	25



## TABLES

---

Table 2-1	Simulation Model Input Parameters . . . . .	10
-----------	---	----

## 1. INTRODUCTION

A simulation model that represents a submarine's search for a high value target within a specified operating area is described in this report. This model was developed as an adjunct to the formulation and implementation of a computationally more efficient analytical model.<sup>\*</sup> The simulation model served two purposes. First, the simulation model provided a validation of the statistical inputs used for the analytical model. Specifically, the simulation studies validated the applicability of the analytical model for determining rate of encounter between submarine and targets. Second, results obtained through exercise of the simulation model provided a convenient check of the reasonableness of analytical model results.

---

<sup>\*</sup> J. M. Moore; "Semi-Markov Models of Search in the Presence of Decoys," NWRC RM-64, SRI Project J016-245, Contract N00014-71-C-0119; Stanford Research Institute, Menlo Park, California; July 1971 (UNCLASSIFIED)

## 2. GENERAL SIMULATION MODEL CHARACTERISTICS

The overall simulation model is comprised of three separate computer submodels. Each of these submodels is designed to represent a specific set of operational conditions which may characterize the various phases of submarine search for a high value target (HVT) in a specified area. The operational conditions represented in these submodels include continuous submarine search--with and without false targets (decoys) present, and submarine search employing the sprint-drift tactic without false targets present.

Each of these submodels is programmed in the BASIC algebraic language for processing and operation by a time-share operating system incorporating a CDC-6400 computer. Because of time-share system limitations, particularly with regard to output of results, these submodels are constructed so as to facilitate independent operations.

Those model characteristics which are common to all three submodels are described and discussed in this section; whereas peculiarities of each submodel are discussed in the sections which follow.

The primary output of each submodel is the elapsed time to HVT detection by the submarine searcher and various statistical quantities which characterize the distribution of the time to HVT detection. Specific

aspects of the various common model characteristics are discussed in the paragraphs which follow.

## 2.1 Operational Framework

In every case, the area is considered to be circular in shape with a known radius,  $R$ . While the radius of the area may be inputted as a parameter, it has usually been assigned a value of 200 nmi.

For reference purposes, the center of the area is made to coincide with the origin of a cartesian coordinate system, oriented so as to position the X-axis in a horizontal direction. Initial and subsequent participant positions are expressed in terms of the underlying coordinate system.

The initial positions of the HVT and decoys (if included) are randomly distributed within the area. This is accomplished through the use of random numbers uniformly distributed on the interval  $[0,1]$  in the following manner.

Let  $u_1$  and  $u_2$  be two numbers drawn randomly from a population of numbers which is uniformly distributed on the interval  $[0,1]$ . Then, for example, the initial position of the HVT,  $P_{co}$ , is determined as:

$$P_{co} = (Ru_2, 2\pi u_1) = [Ru_2 \cos(2\pi u_1), Ru_2 \sin(2\pi u_1)]$$

Initial positions for the decoys are determined in a similar manner utilizing, of course, different random numbers.

In the case of the submarine searcher, the model user may exercise an option to select either an initial position inside the area or a position on the boundary of the area. In this latter case, the initial submarine position,  $P_{so}$ , would be ( $u_3$  is again a random number):

$$P_{so} = (R, 2\pi u_3) = [R \cos(2\pi u_3), R \sin(2\pi u_3)]$$

In the former case, the procedure would be similar to that described above for determining the initial position of the HVT.

For the majority of the cases investigated, an initial submarine position on the boundary of the search area was selected. This condition was considered to more accurately describe the actual operational situations which might be encountered.

## 2.2 Model Time

Each of the submodels employs a fixed time-step form of operation to record elapsed program time. Thus, participant motion and the occurrence/non-occurrence of various events are evaluated at constant, discrete points in time. This form of model construction necessitates a tradeoff between the size of the time-step and the concurrent actual running time of the program. Currently, a time-step of 0.5 hours is used. This interval provides a sufficiently detailed description of the interactions of the various forces, while not requiring exorbitant program running time. Of course, the size of the time-step may be inputted prior to program utilization.

### 2.3 Submarine/HVT/Decoy Motion

Once the initial positions of the various units are determined, the motion of each is the result of a pseudo-random process. That is, the choice of direction is purely random, but the unit speed and length of time on each leg are predetermined. Further, because of the finite nature of the boundary of the area as well as other operational considerations, the motion of the various units may be inhibited to conform to these conditions. Area boundary effects are discussed in this section; other operational factors which inhibit unit motion are discussed in the appropriate sections which follow.

A typical search initiation situation is illustrated in Figure 2-1. Where, as previously described,

$$P_{co} = (Ru_2, 2\pi u_1)$$

and

$$P_{so} = (R, 2\pi u_3)$$

The initial direction of HVT motion is then the angle  $\theta_{co} = 2\pi u_4$ , that is, the initial HVT movement is radially away from the center of the area. This movement is indicated by the dashed line originating from  $P_{co}$  in Figure 2-1. The searcher, on the other hand, moves initially toward the center of the area in a direction equal to  $\theta_{so} = 2\pi u_5$ . It should be noted that the angles  $\theta_{co}$  and  $\theta_{so}$  are measured counterclockwise from the positive X-axis and do not, therefore, correspond to the usual East West, North-South orientation. Decoys, if not stationary, move initially in a manner similar to that exhibited by the HVT. No further

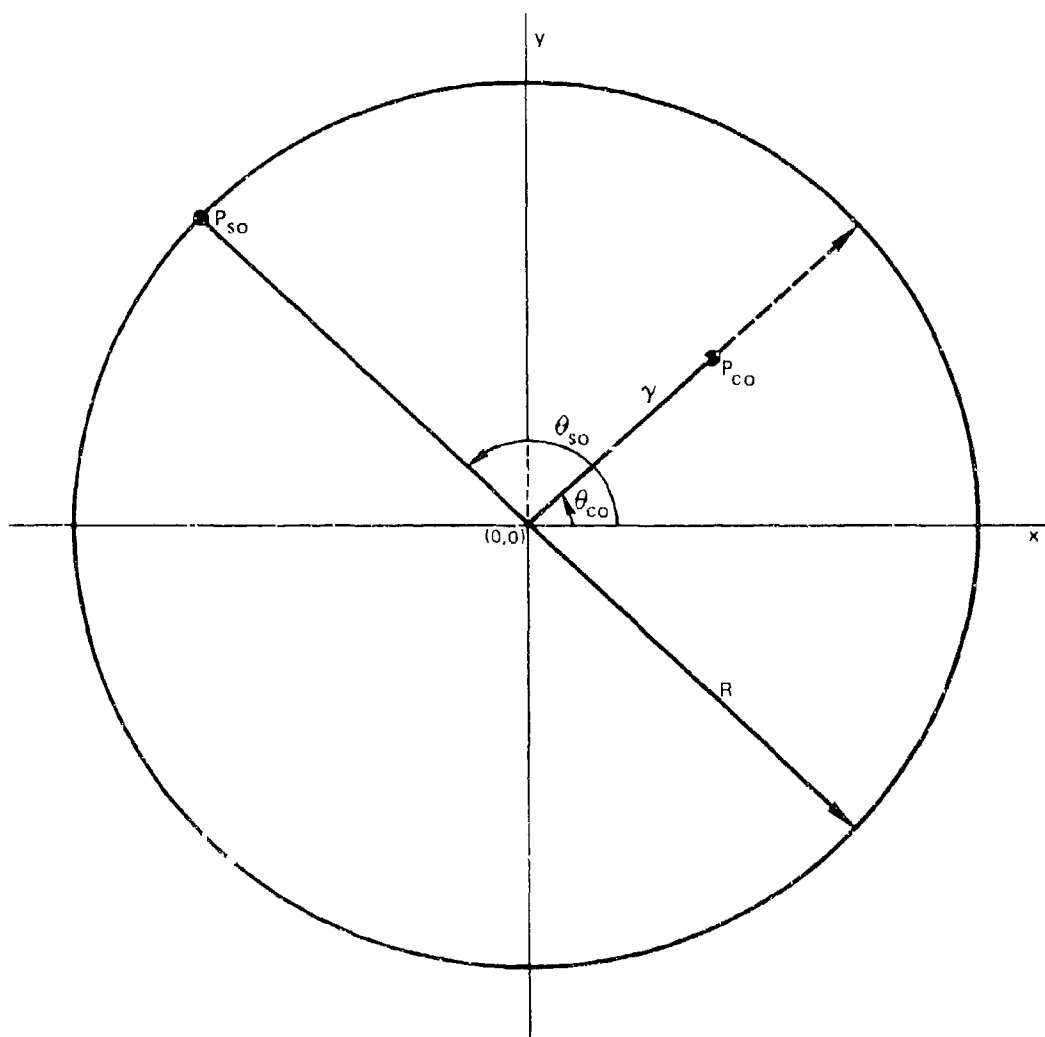


FIGURE 2-1 TYPICAL SEARCH INITIATION SITUATION

comment will be made in this section regarding subsequent decoy movement, for this aspect of the model formulation will be discussed in the section dealing with the peculiarities associated with search in the decoy field environment.

Control over the movement of the HVT and the submarine is exercised in two ways: by specifying, as a program input, the length of each track segment; and automatically, by "reflecting" either the HVT or the submarine off the boundary of the area.

As an illustration of these two procedures, consider that the length of the HVT track segments has been specified (inputted) as a maximum of  $L_c$  nmi. Then, using the situation depicted in the previous diagram, if  $y + L_c \leq R$ , the HVT will move along the dashed line until either a distance  $L_c$  from  $P_{co}$  is reached or the HVT is detected by the submarine.

For either condition: HVT completes movement of  $L_c$  nmi or HVT at the boundary of the area, a new direction of movement is determined by calculating a "turning angle,"  $\theta'$ , such that

$$\theta' = 2\pi u \quad (u \text{ a random number})$$

and the new direction of movement is then

$$\theta_{\text{new}} = \theta_{\text{old}} + \theta'$$

subject to the constraint that, if the HVT is outside the area,  $\theta_{\text{new}}$  must result in returning the HVT to a position within the area.



It should be apparent from the foregoing discussion that the speed of the HVT (and the submarine) is specified prior to beginning the model run and is constant throughout the run.

Movement of the submarine is accomplished in a manner exactly similar to that described for the HVT.

It should be noted that the effect of keeping the submarine within the area boundary reflects an implicit assumption that the submarine possesses perfect information concerning the size and location of the search area. If this were not the case, it would be necessary to overlay the circular area with an area representative of the degree of submarine intelligence information. Such a submarine area might be larger, smaller, and/or offset from the actual area.

#### 2.4 Detection of the HVT and/or Decoy

In each of the submodels, the detection capability of the submarine against either the HVT or the decoy is described by a definite range probability law or "cookie cutter." That is, for some range  $R_D$ , the probability of detection is

$$p [\text{Det}] = \begin{cases} 1 & r \leq R_D \\ 0 & r > R_D \end{cases}$$

where  $r$  is the range between the submarine and either the HVT or the decoy at the end of a particular time-step. In this regard, it is possible for the submarine to pass the HVT or decoy within detection range during

a time-step without a detection being recorded if the range between the final positions is sufficiently large. Again, judicious selection of the time-step, keeping the possible relative speeds in mind, can serve to minimize this possibility.

In actual operation, a value of  $R_D$  may be specified for both the HVT and the decoy which reflects characteristics of the HVT and decoy, the various environmental effects, and the other factors which determine detectability of HVT and decoy. Obviously, one method of reflecting the efficiency of the decoy in accurately impersonating the HVT is to make the detection range of the decoy approach that of the HVT.

## 2.5 Input Parameters

In a slight departure from the previous format of presenting only the common features of the various submodels, all of the necessary input parameters for each of the submodels are presented in Table 2-1. The entries in Table 2-1 are so arranged as to present the common inputs required and then the specific inputs required for each submodel. For ease of reference, the BASIC parameter symbology for the various input quantities is indicated in Table 2-1 rather than the more traditional mathematical notations.

Because of the random aspects introduced into the models, the results from each of the submodels are amenable only to statistical interpretation. For this reason, it is necessary to replicate the model results many times

Table 2-1  
SIMULATION MODEL INPUT PARAMETERS

Basic Symbol	Definition	Units
<u>Common Parameters</u>		
VC	HVT Velocity	kt
COURC	Length of HVT Track Segment ( $COURC \leq R$ )	nmi
R	Radius of Objective Area	nmi
DT	Time Step Increment	hr
NREPLI	Number of Program Replications (Note: NREPLI $\leq$ 500 for Continuous Search--With False Targets Submodel)	Integer
<u>Continuous Search--Without False Targets</u>		
RO	Radius of Detection of HVT by Submarine	nmi
VS	Submarine Search Velocity	kt
COURS	Length of Submarine Track Segment ( $COURS \leq R$ )	nmi
<u>Sprint/Drift Search--Without False Targets</u>		
VSSP	Submarine Sprint Velocity ( $VSSP \cdot SPP \leq R$ )	kt
VSDF	Submarine Drift Velocity ( $VSDF \cdot DFP \leq R$ )	kt
SPP	Submarine Sprint Period	hr
DFP	Submarine Drift Period	hr
RDS	Radius of Detection of HVT by Submarine During Sprint	nmi
RDD	Radius of Detection of HVT by Submarine During Drift	nmi
LFILE	Program Restart Parameters--for Use in Case of Program Execution Interruption. Initial Values: LFILE = 1, LDEC = 0, IP = 1	Integer
LDEC		Integer
IP		Integer
<u>Continuous Search--With False Target Field</u>		
VS	Submarine Search Velocity	kt
COURS	Length of Submarine Track Segment ( $COURS \leq R$ )	nmi
COURD(I)	Length of Decoy Track Segment [ $I = 1, 2, 3, 4, 5$ ] [ $COURD(I) \leq R$ ]	nmi
VDC(I)	Decoy Velocity [ $I = 1, 2, 3, 4, 5$ ] [ $VDC(I) > 0$ ]	kt
TC	Decoy Classification Time	hr
TM	Decoy "Turned Off" Time	hr
RDR	Radius of Detection of Decoy by Submarine	nmi
RZR	Radius of Detection of HVT by Submarine	nmi

for each set of operational conditions which are to be investigated. Thus, provision is included in each submodel for user specification of the number of replications of model results desired. It should be noted that, because of the increased computational time due to the added presence of decoy, the number of replications for the continuous search with the decoy submodel must be limited to less than, or at most, 500.

All other entries in Table 2-1 are either self-explanatory or are discussed elsewhere in this technical note.

## 2.6 Submodel Outputs

In each case, the principal quantity measured is the elapsed time to first detection of the HVT by the submarine. Since detection is characterized by a definite range probability law, first detection is equivalent to first encounter, where encounter occurs whenever the range between the submarine and the HVT is less than some predetermined value.

The output of each submodel consists then of the sample mean, variance, and standard deviation of the time to first detection. For the two continuous search submodels, these values are indicated by the BASIC symbols: MEANC, VARC, and STDEVC, respectively. For the sprint drift submodel, the output format has been modified to print out the full titles: "STANDARD DEVIATION," etc.

In computing the sample variance, the consistent estimator formulation for the population variance is used, that is:

$$S^2 = \frac{1}{M} \sum_{i=1}^M (T_i - \bar{T})^2 ,$$

where  $\bar{T}$  is the sample mean  $\left( = \frac{1}{M} \sum_{i=1}^M T_i \right)$  and  $S^2$  is the sample variance.

The output of the two continuous search submodels also includes both a frequency count and a cumulative frequency of occurrence of first detection as a function of elapsed time interval. Further, the output of the continuous search with the decoy submodel includes a listing of specific times of first encounter of the HVT with the submarine for each of the replications.

### 3. CONTINUOUS SEARCH - NO DECOY SUBMODEL

This submodel represents the basic structure for all of the submodels constructed to date. As such, the internal structure of this submodel is exactly similar to that described in the preceding section. A single submarine searcher seeks a single HVT within a delineated objective operating area. The two units move in a pseudo-random manner, staying within the objective area at all times. Each case or replication is terminated at the instant of initial submarine-HVT encounter/detection. The results obtained from exercising this submodel for ranges of the various input parameters provide the baseline data for evaluating the effectiveness of various tactical procedures, ACM employment policies, and combinations of both.

A typical example of the type of motion which this submodel generates, records, and evaluates is shown in Fig. 3-1.

An example of the results obtained from exercising this submodel is presented in Fig. 3-2.

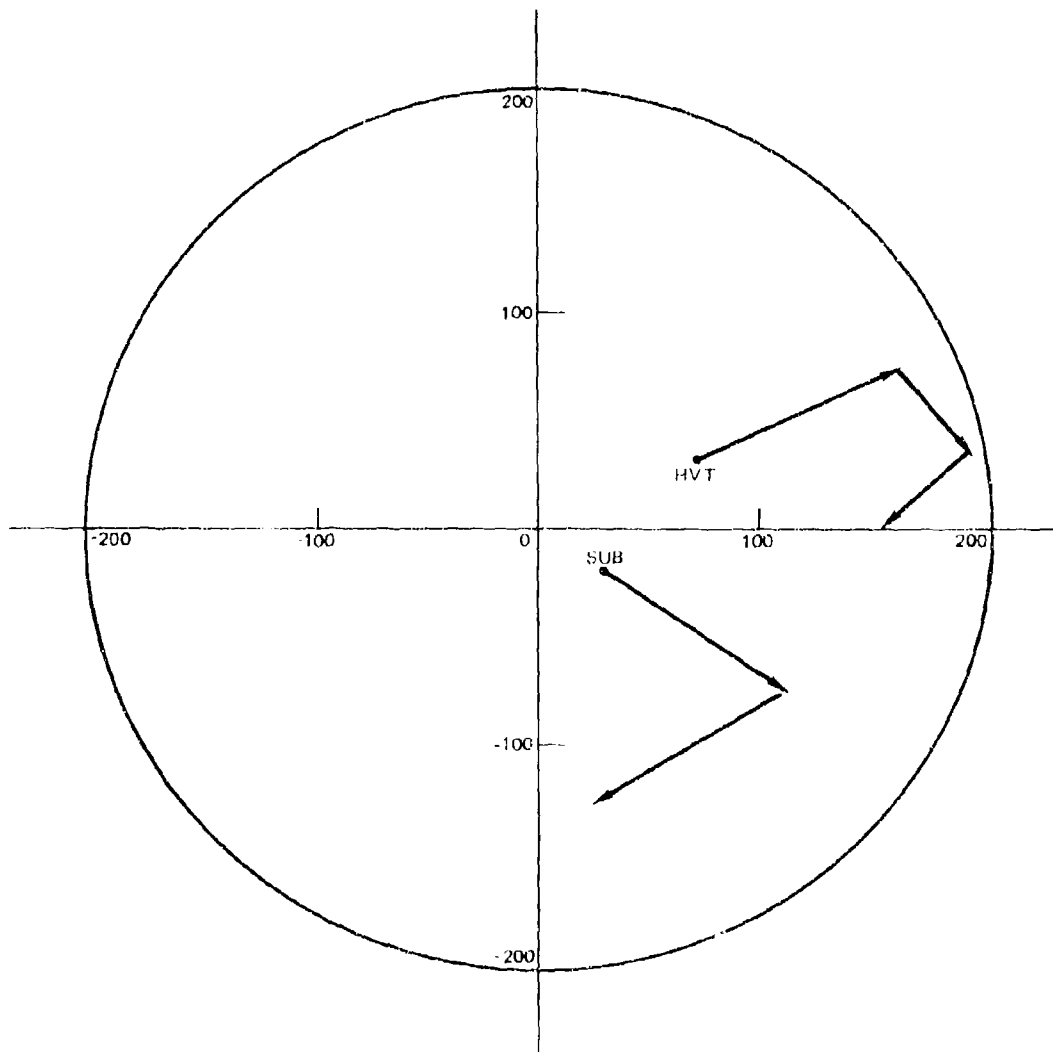


FIGURE 3-1 SAMPLE PLOT OF HVT AND SUBMARINE MOTION

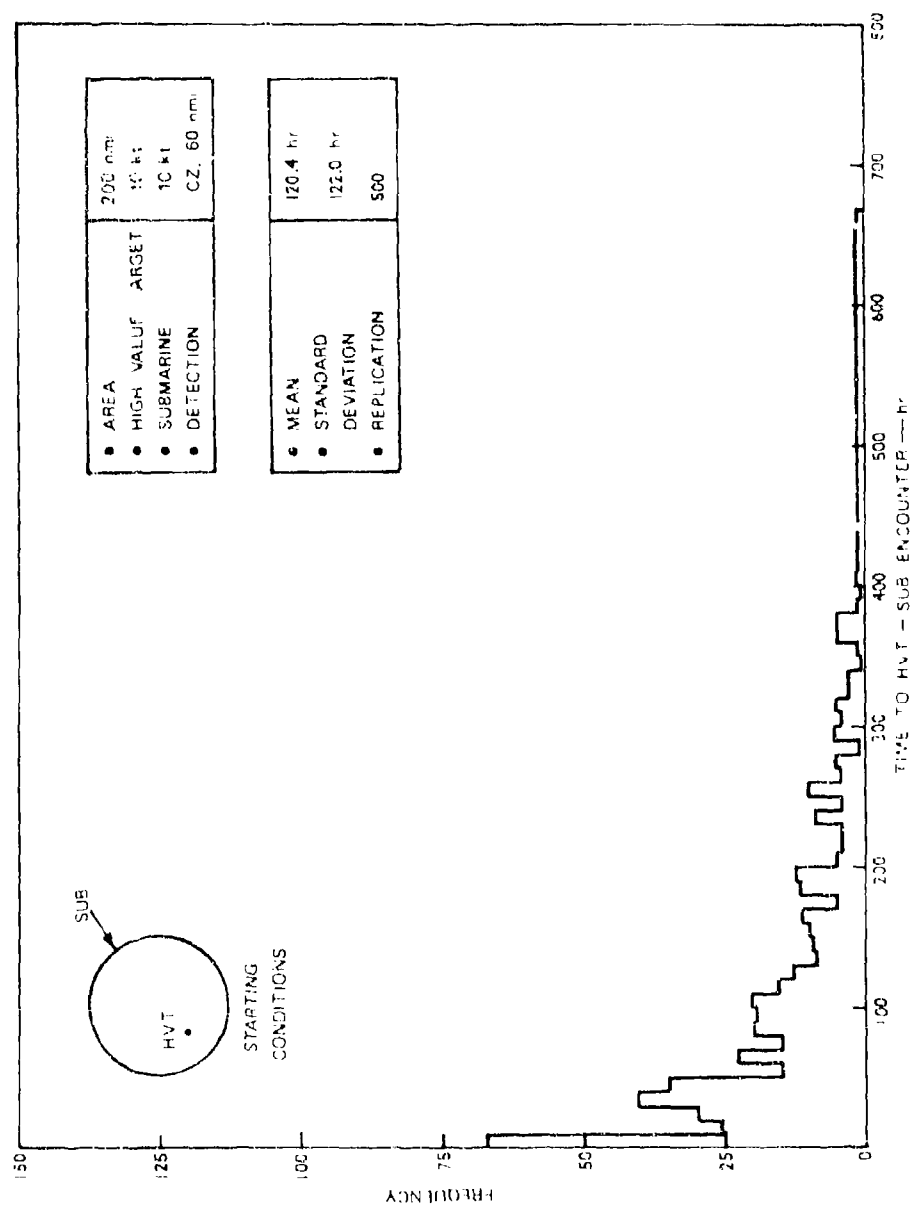


FIGURE 3-2 SAMPLE RESULTS FROM THE CONTINUOUS SEARCH — NO DECOY SUBMODEL



#### 4. SPRINT/DRIFT SEARCH - NO DECOY SUBMODEL

This submodel differs from the baseline submodel in two principal aspects: the manner in which the motion of the submarine is regulated and the manner in which detection of the HVT may occur.

As indicated in the title of this submodel, the submarine employs a tactic of first executing a high speed sprint and then a slow speed drift to attempt detection of the HVT. Thus, motion of the submarine is characterized by four input parameters: the sprint speed, the sprint period, the drift speed, and the drift period. In this manner, the single length of submarine track segment (COURS) specified in the baseline submodel is replaced by two track lengths equal to  $(VSSP \cdot SSP)$  and  $(VSDF \cdot DFP)$ , respectively. (See Table 2-1 for definition of symbols.) At the end of each of these track lengths, new, random headings are determined for the submarine. This procedure is illustrated in the example plot of submarine and HVT movement presented in Figure 4-1.

Under actual conditions, a submarine is effectively "acoustically blind" during the sprint period. There may be instances, however, when the speed selected for the sprint tactic is not so high as to completely eliminate the detection capability of the submarine, especially if the submarine and the HVT should pass close aboard one another. For this reason, provision is included in this submodel for specifying a submarine

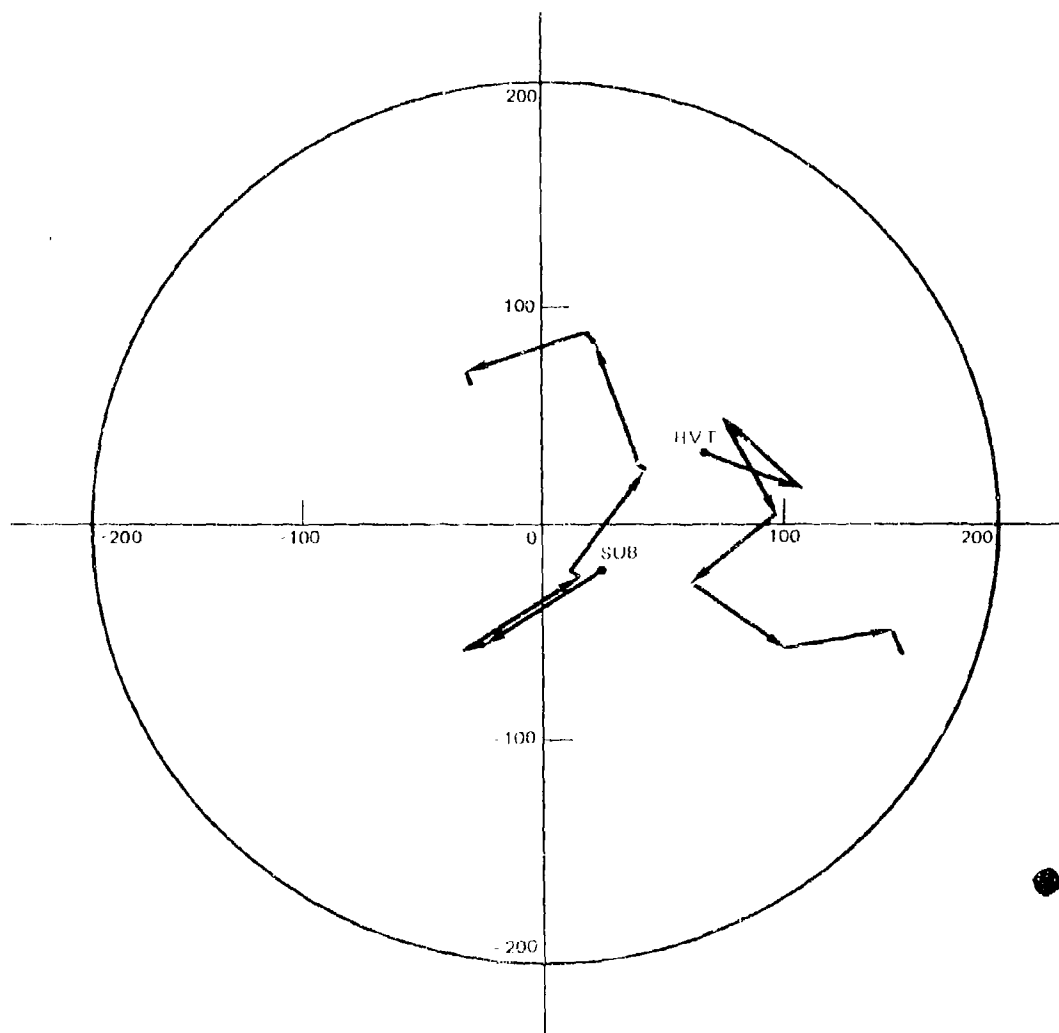


FIGURE 4.1 SAMPLE PLOT OF MOTIONS OF HVT AND SUBMARINE IN THE SPRINT/DRIFT MODEL

detection capability during the sprint period although, in all runs to date, this value has been set so low as to effectively preclude such detections.

Four programs are required to exercise this submodel. These programs are: "SPRINT," "FILECR," "UPDATE," and "STATICS." The last of these, "STATICS," performs the statistical analysis of the results of the various replications and prints the summary data.

The program, "SPRINT," comprises the main program of this submodel and models all of the details of the search. As such, this program requires specification of all of the initial input parameter values indicated in Table 2-1. Prior to the execution of SPRINT, an independent subroutine, LITBAL, must be called into the processor. This is accomplished by the system command, "Get, LITBAL."

The two programs, "FILECR" and "UPDATE," are utility programs that manipulate permanent program data files. These programs were found to be necessary in order to safeguard against loss of accumulated data in the case of machine failure or program interruption during lengthy program executions. The three parameters, "LFILE," "LREC," and "IP" are used to recover and restart program execution should it be interrupted. (LFILE is the BASIC symbol for "Last File Updated," and LREC is the symbol for "Last Record Created.")

The output from the SPRINT program is printed on magnetic tape after every 20th replication. This output is then available for either

statistical analysis, using the STATIC program, and/or permanent storage on magnetic tape using the UPDATE program.

An example of the results obtained from exercising this submodel is presented in Figure 4-2.

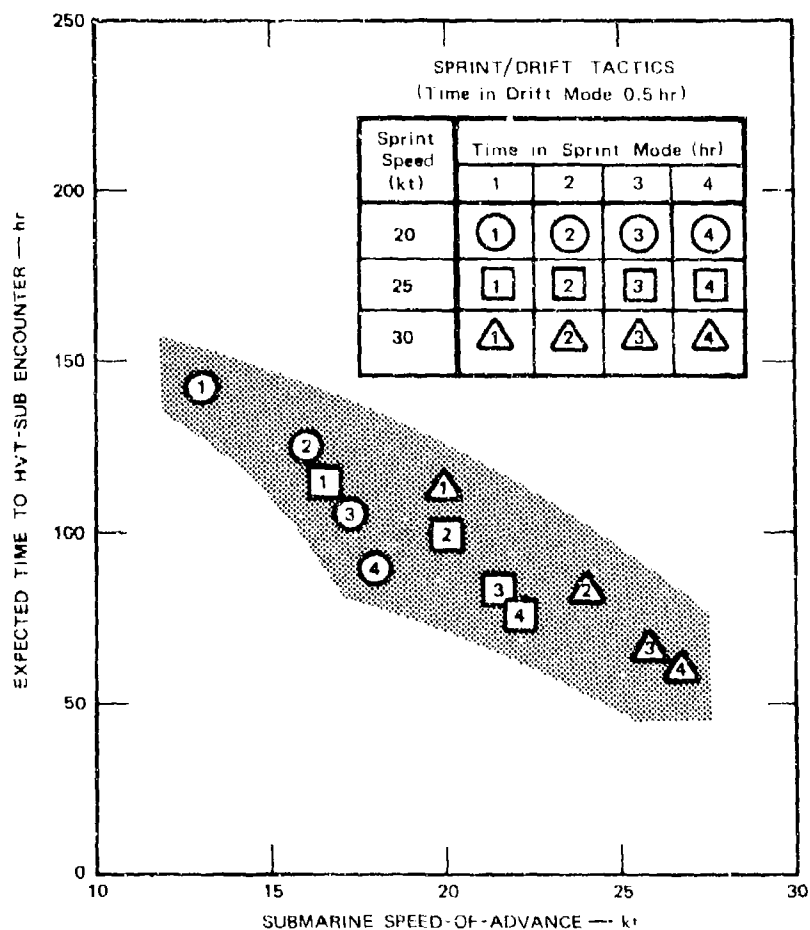


FIGURE 4-2 SAMPLE RESULTS FROM THE SPRINT/DRIFT SEARCH —  
NO DECOY SUBMODEL

## 5. CONTINUOUS SEARCH - DECOY FIELD SUBMODEL

As the title implies, this submodel incorporates the effects on searcher effectiveness due to the presence of decoys within a specified area. This innovation requires several departures from the procedures established for the baseline submodel.

As before, the submodel includes a single submarine searcher and a single HVT. Due to limitations on computer core storage in the present time-share system, the number of decoys must be restricted to five or less. Increased capacity can be achieved through the use of utility programs such as those described in the preceding section, but this has not been done at this time.

Either stationary or moving decoys can be simulated through appropriate specification of the decoy velocity,  $VDC(I)$ . Because of some of the computations involving  $VDC$ , however, this value cannot be zero. Therefore, the stationary decoy is represented by making  $VDC$  very small, albeit, positive.

The HVT and the decoys are initially positioned within the area; the submarine searcher on the boundary. All of the units are permitted to move in the pseudo-random manner described before subject to the provision that no decoy may approach the HVT closer than  $(R1R + R2R)nmi.$  (See Table 2-1 for definition of symbols.) This constraint insures

that the influence fields (detection radii) of the HVT and the decoys shall never overlap. If the decoys are capable of movement, the HVT is allowed to move in an unrestricted manner and the decoys are required to remain clear of the HVT. Obviously, if the decoys are considered to be stationary, i.e., VDC is very small, this condition must be reversed. If overlap should occur at the time of determination of initial positions, the initial positions of the decoys are adjusted so as to eliminate the overlap.

If the submarine searcher comes within detection range of a decoy, the submarine is considered to spend a period equivalent to TC hours classifying the decoy. During this time, the submarine is precluded from making new detections on either the HVT or other decoys. The period TC is also sometimes referred to as "decoy capture time" or decoy hold time."

During the period TC, the decoy continues to move in accordance with the appropriate pseudo-random procedures. The submarine searcher is assumed to move in consonance with the decoy. The position of the decoy at the end of the period TC is taken to be the re-start position of the submarine searcher.

To simulate submarine memory of classified decoy locations, the decoys are "turned off" by the model for a time TM following the classification period, TC. This permits the submarine to clear the immediate

area of the decoy. Obviously, the period TM should be made at least as long as RIR/VS when specifying this input value.

If it should occur that the submarine is within detection range of two or more decoys at any given instant, the model assumes that the submarine will investigate and classify the nearest decoy, ignoring the others. If ever the HVT is within range, the model assumes that it will be detected and classified by the submarine without regard for whatever decoys may also be present. As in all other cases, the instant of first detection of the HVT by the submarine terminates the specific replication.

A typical example of the motion generated for the various units within this submodel is presented in Figure 5-1. In this example, 4 decoys are present within the objective area (D1, D2, D3, and D4 in Figure 5-1) along with the HVT and submarine searcher.

An example of the type of results obtained from exercising this submodel is shown in Figure 5-2.



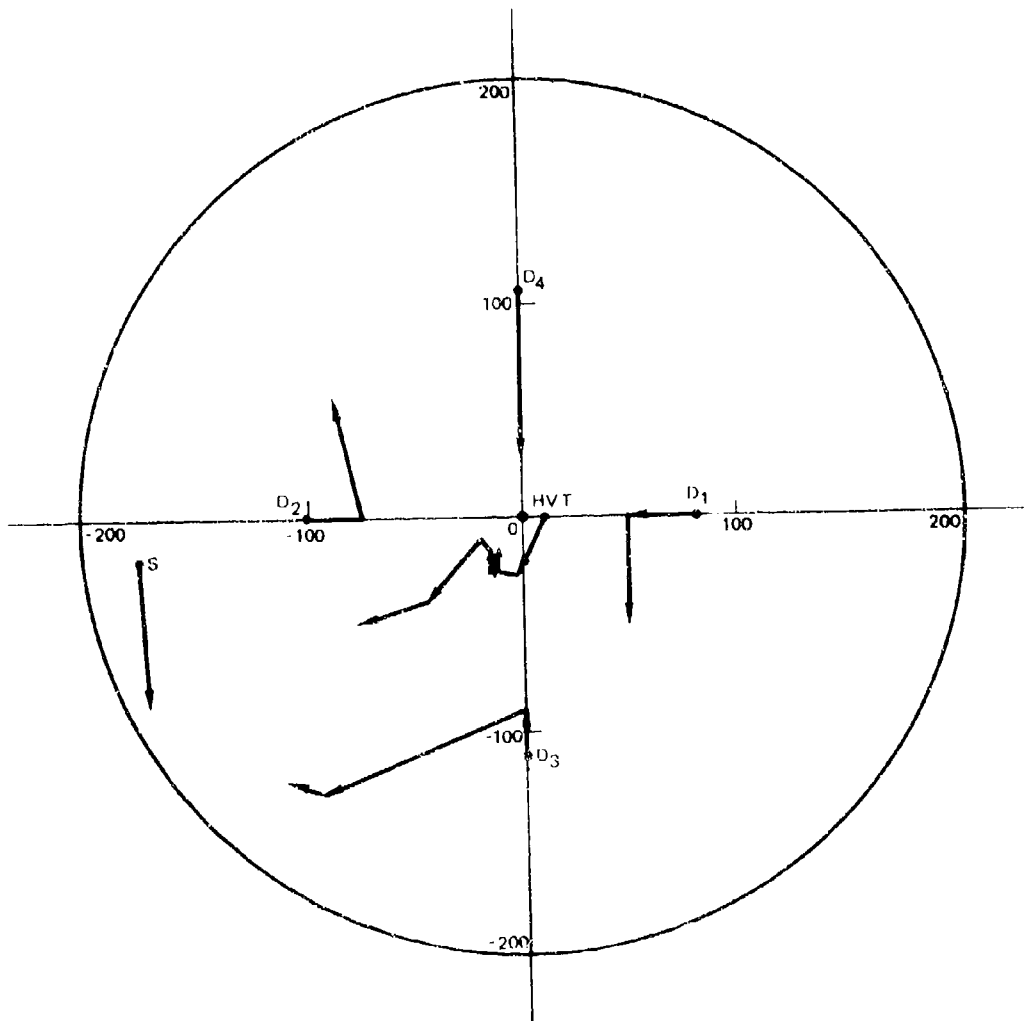


FIGURE 5-1 SAMPLE PLOT OF MOTIONS OF FOUR DECOYS ( $D_1, \dots, D_4$ ), SUBMARINE, AND HVT FROM A TEST RUN WHERE ALL EXCEPT THE SUBMARINE ARE PREPOSITIONED INITIALLY

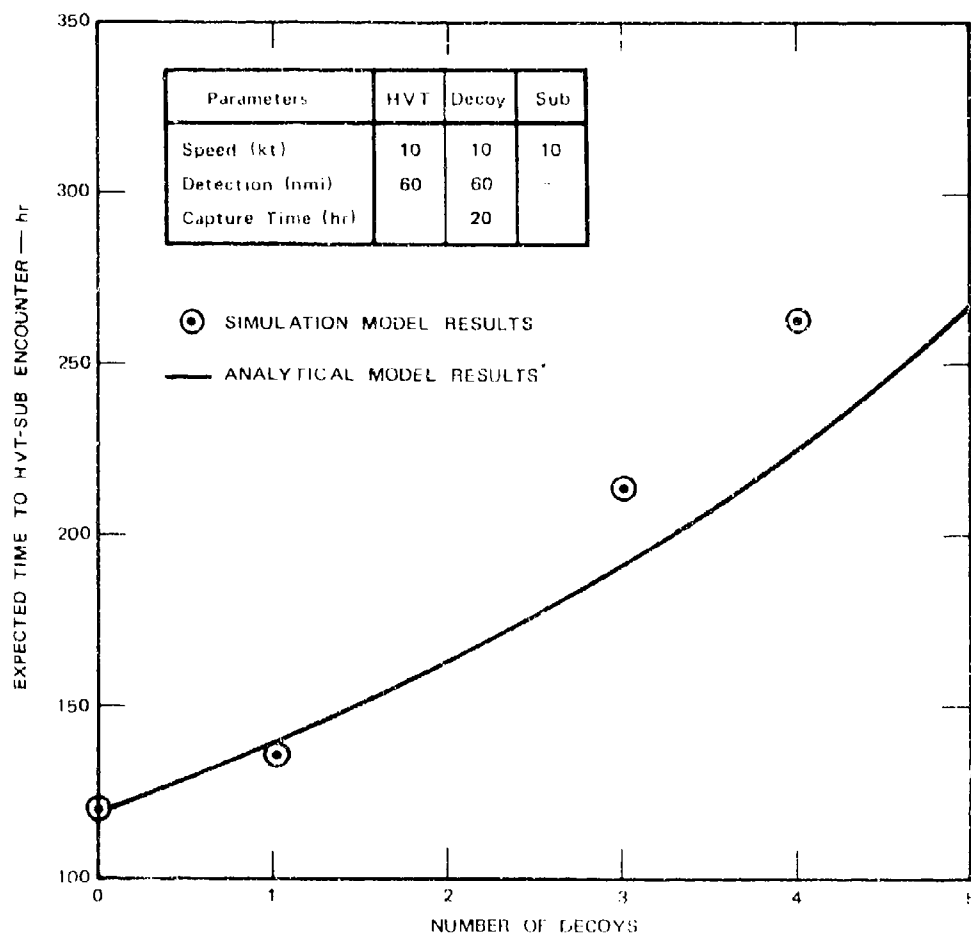


FIGURE 5-2 SAMPLE RESULT\* FROM THE CONTINUOUS SEARCH — DECOY FIELD SUBMODEL

\* J. M. Moore; "Semi-Markov Models of Search in the Presence of Decoys," NWRC RM-64, SRI Project 1016-245, Contract N00014-71-C-0119; Stanford Research Institute, Menlo Park, California; July 1971 (UNCLASSIFIED)

Appendix A

PROGRAM LISTING FOR CONTINUOUS SEARCH -  
NO DECOY SUBMODEL (MOD 1)

MOD1 06/10/71.

```
00100 PROGRAM MOD1(INPUT,OUTPUT)
00110 DIMENSION TND(5),XD(5),YD(5),ANGD(5),IDN(5),COURD(5),
00120+ NOPD(5)
00130 DIMENSION VD(5)
00140 DIMENSION TDCVA(1000)
00150 DIMENSION TDCOY(100)
00160 DATA COURD/5.,5.,5.,5.,5./
00170 DATA VD/1.,1.,1.,1.,1./
00180 R=200.
00190 NINT=500
00200 ITGT5=0
00210 IP1=1
00220 IP2=5
00230 PRINT,*C*
00240 PRINT,*CVA RANDOM START IN AREA*
00250 PRINT,*SUB RANDOM START ON BOUNDARY*
00260 PRINT,*C*
00270 CALL SECOND(X)
00280 Y=RANF(X)
00290 XX=0.
00300 PRINT,*ENTER VC,VS,COURC,COURS,RO*
00310 READ,VC,VS,COURC,COURS,RO
00320 I=0
00330 PRINT,*ENTER NREPLI*
00340 READ,NREPLI
00350 TIME=0.
00360 HRS=0.
00370 NDECOY=0
00380 TDELAY=(-9.*VS)/50.+5.9
00390 KOUNT=0
00400 KOND=0
00410 KONC=0
00420 PRINT,*COUNTER,T ELAPSED,T ENCOUNTERED*
00430 202 CONTINUE
00440 INC=0
00450 INS=0
00460 TNC=0.
00470 TNS=0.
00480 HRS=0.
00490 DFG=57.+(17./60.)
00500 TD=0.
00510 NOPC=0
00520 NOPS=0
00530 DO 2 I=1,NDECOY
00540 NOPD(I)=0
00550 IDN(I)=0
00560 TND(I)=0.
00570 2 CONTINUE
00580 200 CONTINUE
00590 ANGC=6.28319*RANF(XX)
```

MOD1 06/10/71.

```
00600  RADUS=R*RANF(XX)
00610  XC=RADUS*COS(ANGC)
00620  YC=RADUS*SIN(ANGC)
00630  ANG5=6.28319*RANF(XX)
00640  RAD=R
00650  XS=RAD*COS(ANGS)
00660  YS=RAD*SIN(ANGS)
00670  IF(NDECOY.EQ.0) GO TO 800
00680  DO 4 I=1,NDECOY
00690  CALL INDCOY(ANGD(I),XD(I),YD(I),XC,YC,XS,YS,XX,R,VC,VS)
00700  TND(I)=TND(I)+1.
00710  4 CONTINUE
00720  800 CONTINUE
00730  DT=0.5
00740  15 CONTINUE
00750  TIME=TIME+DT
00760  HRS=HRS+DT
00770  CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
00780+ NOPC,TDELAY,TD)
00790  XC=XNOW
00800  YC=YNOW
00810  CALL LITBAL(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,COURS,XX,
00820+ NOPS,TDELAY,TD)
00830  XS=XNOW
00840  YS=YNOW
00850  IF(NOPS.EQ.2) GO TO 360
00860  IF(NDECOY.EQ.0) GO TO 350
00870  DO 350 I=1,NDECOY
00880  CALL LITBAL(TND(I),XD(I),YD(I),ANGD(I),XNOW,YNOW,IND(I),
00890+ VD(I),COURD(I),XX,NOPD(I),TDELAY,TD)
00900  XD(I)=XNOW
00910  YD(I)=YNOW
00920  350 CONTINUE
00930  CALL MEFT(VC,VS,XC,YC,XS,YS,IEN)
00940  IF(IEN.EQ.1) GO TO 100
00950  GO TO 300
00960  100 CONTINUE
00970  KOUNT=KOUNT+1
00980  KONC=KONC+1
00990  TDCVA(KONC)=HRS
01000  IPRINT=IP1*10
01010  IF(IPRINT.NE.KOUNT) GO TO 1000
01020  IP1=IP1+1
01030  PRINT 95,KOUNT,HRS,TIME
01040  95 FORMAT(I10,2F11.4/)
01050  1000 CONTINUE
01060  IF(KOUNT.EQ.NREPLI) GO TO 97
01070  ISTAT=IP2*10
01080  IF(ISTAT.NE.KOUNT) GO TO 1010
01090  CALL TIC(TDCVA,TDCOY,KONC,KOND,EXPT,VAR,ISTAT,VS)
```

MOD1 06/10/71.

```
01100 CALL FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
01110 PRINT,*TIME GT 5000*
01120 PRINT,ITGT5
01130 IP2=IP2+5
01140 GO TO 202
01150 1010 CONTINUE
01160 GO TO 202
01170 300 CONTINUE
01180 IF(NDECOY.EQ.0) GO TO 15
01190 DO 361 I=1,NDECOY
01200 CALL MEET(VC,VS,XD(I),YD(I),XS,YS,IEN)
01210 IF(IEN.EQ.1) GO TO 500
01220 361 CONTINUE
01230 IF(KOUNT.EQ.NREPLI) GO TO 97
01240 GO TO 15
01250 310 CONTINUE
01260 IF(NOPS.EQ.1) GO TO 15
01270 PRINT,*MEET DECOY*
01280 PRINT,I,XD(I),YD(I)
01290 PRINT,XS,YS
01300 IKFFP=I
01310 NOPS=1
01320 NOPD(I)=2
01330 ID=0.
01340 GO TO 15
01350 360 CONTINUE
01360 NOPS=0
01370 NOPD(IKEEP)=0
01380 XS=XD(IKEEP)
01390 YS=YD(IKEEP)
01400 DO 370 I=1,NDECOY
01410 IF(I.EQ.IKEEP) GO TO 370
01420 CALL MEET(VC,VS,XD(I),YD(I),XS,YS,IEN)
01430 IF(IEN.EQ.1) GO TO 310
01440 370 CONTINUE
01450 IKFFP=0
01460 GO TO 15
01470 99 CONTINUE
01480 500 PRINT,*DECOY ENCOUNTERED,T ENCOUNTERED*
01490 PRINT,I,HRS
01500 98 CONTINUE
01510 KOND=KOND+1
01520 TDCOY(KOND)=HRS
01530 KOUNT=KOUNT+1
01540 GO TO 202
01550 97 CONTINUE
01560 IF(KOND.EQ.0) GO TO 504
01570 504 CONTINUE
01580 CALL TIC(TDCVA,TDCOY,KONC,KOND,FXPT,VAR,NREPLI,VS)
01590 CALL FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
```

MOD1 06/10/71.

01600 PRINT,\*FREQUENCY GT 5000\*  
01610 PRINT,ITGT5  
01620 END

MOD1      06/10/71.

```
01630  SUBROUTINE  STAT(TIME, ICOUT, FEAN, STDEV, VAR)
01640  DIMENSION  TIME(1000)
01650  SUM=0.
01660  DO  5  I=1, ICOUT
01670  SUM=SUM+TIME(I)
01680  5  CONTINUE
01690  FEAN=SUM/FLOAT(ICOUT)
01700  VAR=0.
01710  DO  10  I=1, ICOUT
01720  VAR=VAR+(TIME(I)-FEAN)**2
01730  10  CONTINUE
01740  VAR=VAR/FLOAT(ICOUT)
01750  STDEV=SQRT(VAR)
01760  RETURN
01770  END
```



MOD1      06/10/71.

```
01780  SUBROUTINE TIC(TDCVA,TDCOY,KONC,KOND,EXPT,VAR,NREPLI,VS)
01790  DIMENSION TDCVA(1000),TDCOY(100)
01800  PRINT,*NO OF REPLICATIONS*
01810  PRINT,NREPLI
01820  TDLAY=(-9.*VS)/50.+5.9
01830  IF(KOND.EQ.0) GO TO 10
01840  CALL STAT(TDCOY,KOND,MEAND,STDEV,VAR)
01850  PRINT,MEAND,STDEV,VAR
01860  10 CONTINUE
01870  PRINT,*MEANC,STDEV,VARC*
01880  CALL STAT(TDCVA,KONC,FEANC,STDEV,VARC)
01890  PRINT,FEANC,STDEV,VARC
01900  IF(KOND.EQ.0) RETURN
01910  EXPT=(P/(1.-P))*(TDLAY+MEAND)+TDLAY+FEANC
01920  TERM1=P/((1.-P)**2)
01930  TERM2=(TDLAY+MEAND)**2
01940  TERM3=(P/(1.-P))*VAR
01950  PRINT,TERM1,TERM2,TERM3
01960  VAR=TERM1*TERM2+TERM3+VARC
01970  RETURN
01980  END
```

MODI 06/10/71.

```
01990 SUBROUTINE LITRAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,
02000+ COURSE,XX,NOP,TDELAY,TD)
02010 IF(NOP.EQ.1) GO TO 30
02020 DT=.5
02030 R=200.
02040 IF(IND.EQ.0) GO TO 20
02050 FLEN=TN*DT*V
02060 IF(FLEN.GT.COURSE) GO TO 20
02070 DX=(DT*V)*COS(ALAST)
02080 DY=(DT*V)*SIN(ALAST)
02090 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
02100 IF(RNOW.GT.R) GO TO 20
02110 XNOW=XLAST+DX
02120 YNOW=YLAST+DY
02130 GO TO 10
02140 30 CONTINUE
02150 XNOW=XLAST
02160 YNOW=YLAST
02170 IF(TD.LE.TDELAY) GO TO 15
02180 PRINT,*LAST TD*
02190 PRINT,TD
02200 IEN=0
02210 NOP=2
02220 TD=0.
02230 ALAST=6.28319*RANF(XX)
02240 IND=1
02250 TN=1.
02260 PRINT,*T DELAY END*
02270 GO TO 15
02280 15 TD=TD+DT
02290 PRINT,*TD*
02300 PRINT,TD
02310 GO TO 25
02320 20 CONTINUE
02330 IND=1
02340 TN=1.
02350 ALPHA=6.28319*RANF(XX)
02360 PHY=ALPHA+ALAST
02370 PHY=AMOD(PHY,6.28319)
02380 DX=(DT*V)*COS(PHY)
02390 DY=(DT*V)*SIN(PHY)
02400 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
02410 IF(RNOW.GT.R) GO TO 20
02420 ALAST=PHY
02430 XNOW=XLAST+DX
02440 YNOW=YLAST+DY
02450 10 TN=TN+1.
02460 25 CONTINUE
02470 RETURN
02480 END
```

MOD1 06/10/71.

```
02490 SUBROUTINE INDCOY(ANGD1,XD1,YD1,XC,YC,XS,YS,XX,R,VC,VS)
02500 5 CONTINUE
02510 ANGD1=6.28319*RANF(XX)
02520 XD1=R*COS(ANGD1)
02530 YD1=R*SIN(ANGD1)
02540 CALL MEET(VC,VS,XD1,YD1,XS,YS,IEN)
02550 IF(IEN.EQ.1) GO TO 5
02560 RETURN
02570 END
```

MOD1      06/10/71.

```
02580  SUBROUTINE  MEET(VC,VS,XC,YC,XS,YS,IEN)
02590  R0=60.
02600  RANGE=SQRT((XC-XS)**2+(YC-YS)**2)
02610  IF(RANGE.GT.R0)  GO  TO  10
02620  IEN=1
02630  GO  TO  20
02640  10  IEN=0
02650  20  CONTINUE
02660  RETURN
02670  END
```

MOD1 06/40/71.

```
02680 SUBROUTINE FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGTS)
02690 DIMENSION TDCVA(1000),IFRE(500),TCOM(500)
02700 DIMENSION IFREC(500)
02710 ITGT5=0
02720 IF(ISTAT.LT.NREPLI) GO TO 5
02730 ISUM=NREPLI
02740 GO TO 6
02750 5 ISUM=ISTAT
02760 6 TINT=10.
02770 TMAX=0.
02780 DO 7 I=1,NINT
02790 IFRE(J)=0
02800 IFREC(J)=0
02810 7 CONTINUE
02820 DO 10 I=1,ISUM
02830 DO 20 J=1,NINT
02840 TCOMPA=FLOAT(J)*TINT
02850 IF(TDCVA(I).GT.5000.) GO TO 40
02860 IF(TDCVA(I).GT.TCOMPA) GO TO 20
02870 IFRE(J)=IFRE(J)+1
02880 IF(TCOMPA.LT.TMAX) GO TO 10
02890 TMAX=TCOMPA
02900 GO TO 10
02910 40 ITGT5=ITGT5+1
02920 GO TO 10
02930 20 CONTINUE
02940 10 CONTINUE
02950 IP=TMAX/TINT+1
02960 DO 30 I=1,IP
02970 TCOM(I)=FLOAT(I)*TINT
02980 30 CONTINUE
02990 IFRE(1)=IFRE(1)
03000 DO 110 I=2,IP
03010 IFRE(I)=IFRE(I)+IFRE(I-1)
03020 110 CONTINUE
03030 PRINT,*TIME INTERVAL,FREQUENCY*
03040 DO 100 I=1,IP
03050 IF(IFRE(I).EQ.0) GO TO 100
03060 PRINT 35,TCOM(I),IFRE(I),IFREC(I)
03070 35 FORMAT(1H,2X,E11.2,2I10)
03080 100 CONTINUE
03090 RETURN
03100 END
```

Appendix B

PROGRAM LISTING FOR SPRINT/DRIFT SEARCH -  
NO DECOY SUBMODEL (SPRINT)

SPRINT 06/10/71.

```
00100 PROGRAM SPRINT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3)
00110 CALL RETR(1,5HTAPE1)
00120 CALL RETR(2,5HTAPE2)
00130 REWIND 3
00140 USE(LITBAL)
00150 PRINT,*LAST FILE UPDATE?*
00160 READ,LFILE
00170 PRINT,*LAST REC CREATED?*
00180 READ,LREC
00190 PRINT,*VSSP,RDS,SPP?*
00200 READ,VSSP,RDS,SPP
00210 PRINT,*VSDF,RDD,DFP?*
00220 READ,VSDF,RDD,DFP
00230 PRINT,*SUB RANDOM START IN AREA=1,ON BOUNDARY=2,WHICH?*
00240 READ,ISTART
00250 R=200.
00260 PRINT,*NO OF REPLICATIONS?*
00270 READ,NREPLI
00280 PRINT,*VC,COURC?*
00290 READ,VC,COURC
00300 XX=0.
00310 PRINT,*ENTER IP*
00320 READ,IP
00330 ICOPY=IP*20
00340 K=LFILE
00350 KOUNT=LREC
00360 CALL SECOND(X)
00370 Y=RANF(X)
00380 220 CONTINUE
00390 ANGC=6.28319*RANF(XX)
00400 RADUS=R*RANF(XX)
00410 XC=RADUS*COS(ANGC)
00420 YC=RADUS*SIN(ANGC)
00430 ANGS=6.28319*RANF(XX)
00440 GO TO (10,20),ISTART
00450 10 RADUS=R*RANF(XX)
00460 GO TO 25
00470 20 RADUS=R
00480 25 CONTINUE
00490 XS=RADUS*COS(ANGS)
00500 YS=RADUS*SIN(ANGS)
00510 DT=.5
00520 TIME=0.
00530 INC=0
00540 INS=0
00550 INC=0.
00560 TNS=0.
00570 Y=RANF(XX)
00580 ISPSW=0
00590 IF(Y.GT..5) ISPSW=1
```

SPRINT 06/10/71.

```
00600 CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
00610+ DT,R)
00620 XC=XNOW
00630 YC=YNOW
00640 IF(ISPSW.EQ.1) GO TO 30
00650 TMSP=TIME+DFP
00660 VS=VSDF
00670 COURS=VSDF*DFP
00680 RD=RDD
00690 GO TO 35
00700 30 TMSP=TIME+SPP
00710 VS=VSPP
00720 COURS=VSPP*SSP
00730 RD=RDS
00740 35 INS=0
00750 TNS=0.
00760 150 CONTINUE
00770 CALL MOV5(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,COURS,XX,
00780+ DT,R)
00790 XS=XNOW
00800 YS=YNOW
00810 DIST=SQRT((XC-XS)**2+(YC-YS)**2)
00820 IF(DIST.LE.RD) GO TO 100
00830 IF(TMSP.EQ.TIME) GO TO 110
00840 IF(TMSP.GT.TIME) GO TO 120
00850 PRINT,*ERR1-TMSP.LE.TIME*
00860 110 CONTINUE
00870 IF(ISPSW.EQ.1) GO TO 130
00880 ISPSW=1
00890 TMSP=TIME+SPP
00900 VS=VSPP
00910 RD=RDS
00920 GO TO 140
00930 130 CONTINUE
00940 ISPSW=0
00950 TMSP=TIME+DFP
00960 VS=VSDF
00970 COURS=VSDF*DFP
00980 RD=RDD
00990 140 INS=0
01000 TNS=0.
01010 12 TIME=TIME+DT
01020 CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,VC,COURC,XX,
01030+ DT,R)
01040 XC=XNOW
01050 YC=YNOW
01060 GO TO 150
01070 100 CONTINUE
01080 KOUNT=KOUNT+1
01090 WRITE(3,160) KOUNT,TIME
```



SPRINT 06/11/71.

```
01100 160 FORMAT(110,E20.4)
01110 IF(KOUNT.LT.ICOPY) GO TO 220
01120 IF(IREC.NE.0) GO TO 404
01130 REWIND 1
01140 REWIND 3
01150 WRITE(1,413) KOUNT
01160 407 READ(3,160) I,TIME
01170 IF(I.EQ.KOUNT) GO TO 406
01180 WRITE(1,160) I,TIME
01190 GO TO 407
01200 406 WRITE(1,160) I,TIME
01210 ENDFILE 1
01220 REWIND 1
01230 CALL REPL(1,5HTAPE1)
01240 REWIND 3
01250 IREC=KOUNT
01260 K=1
01270 GO TO 310
01280 404 CONTINUE
01290 GO TO (400,410) K
01300 400 CONTINUE
01310 REWIND 1
01320 REWIND 2
01330 REWIND 3
01340 WRITE(2,413) KOUNT
01350 READ(1,413) NREC
01360 413 FORMAT(110)
01370 IREC=0
01380 IF(NREC.EQ.0) GO TO 600
01390 415 READ(1,160) I,TIME
01400 IF(I.EQ.NREC) GO TO 411
01410 IREC=IREC+1
01420 WRITE(2,160) IREC,TIME
01430 GO TO 415
01440 411 IREC=IREC+1
01450 WRITE(2,160) IREC,TIME
01460 600 READ(3,160) I,TIME
01470 IF(IREC.EQ.KOUNT) GO TO 420
01480 IF(I.EQ.KOUNT) GO TO 420
01490 IREC=IREC+1
01500 WRITE(2,160) IREC,TIME
01510 GO TO 600
01520 420 IREC=IREC+1
01530 WRITE(2,160) IREC,TIME
01540 ENDFILE 2
01550 REWIND 2
01560 CALL REPL(2,5HTAPE2)
01570 K=2
01580 REWIND 1
01590 REWIND 2
```

SPRINT 06/10/71.

```
01600 REWIND 3
01610 GO TO 310
01620 410 CONTINUE
01630 REWIND 1
01640 REWIND 2
01650 REWIND 3
01660 WRITE(1,413) KOUNT
01670 READ(2,413) NREC
01680 IREC=0
01690 IF(NREC.EQ.0) GO TO 605
01700 610 READ(2,160) I,TIME
01710 IF(I.EQ.NREC) GO TO 615
01720 IREC=IREC+1
01730 WRITE(1,160) IREC,TIME
01740 GO TO 610
01750 615 IREC=IREC+1
01760 WRITE(1,160) IREC,TIME
01770 605 READ(3,160) I,TIME
01780 IF(IREC.EQ.KOUNT) GO TO 620
01790 IF(I.EQ.KOUNT) GO TO 620
01800 IREC=IREC+1
01810 WRITE(1,160) IREC,TIME
01820 GO TO 605
01830 620 IREC=IREC+1
01840 WRITE(1,160) IREC,TIME
01850 ENDFILE 1
01860 REWIND 1
01870 CALL REFL(1,SHTAPE1)
01880 K=1
01890 REWIND 1
01900 REWIND 2
01910 REWIND 3
01920 GO TO 310
01930 310 PRINT,*LAST REC CREATED, LAST FILE CREATED*
01940 PRINT,KOUNT,K
01950 IF(KOUNT.EQ.NREPL1) GO TO 300
01960 IP=IP+1
01970 PRINT,*NEXT IP IS*
01980 PRINT,IP
01990 ICOPY=IP*20
02000 GO TO 220
02010 300 CONTINUE
02020 IP=IP+1
02030 PRINT,*NEXT IP IS*
02040 PRINT,IP
02050 END
```

SPRINT 06/10/71.

```
02060 SUBROUTINE MOV5(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,  
02070+ COURSE,XX,DT,R)  
02080 IF(IND.EQ.0) GO TO 20  
02090 DX=DT*V*COS(ALAST)  
02100 DY=DT*V*SIN(ALAST)  
02110 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)  
02120 IF(RNOW.GT.R) GO TO 20  
02130 XNOW=XLAST+DX  
02140 YNOW=YLAST+DY  
02150 GO TO 10  
02160 20 CONTINUE  
02170 IND=1  
02180 ALPHA=6.28319*RANF(XX)  
02190 PHY=ALPHA+ALAST  
02200 PHY=AMOD(PHY,6.28319)  
02210 DX=DT*V*COS(PHY)  
02220 DY=DT*V*SIN(PHY)  
02230 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)  
02240 IF(RNOW.GT.R) GO TO 20  
02250 ALAST=PHY  
02260 XNOW=XLAST+DX  
02270 YNOW=YLAST+DY  
02280 10 CONTINUE  
02290 TN=TN+1.  
02300 RETURN  
02310 END
```

LITBAL 06/10/71.

```
00100+ SUBROUTINE LITBAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,V,COURSE,XX
00120 IF(IND.EQ.0) GO TO 20
00130 FLEN=TN*DT*V
00140 IF(FLEN.GT.COURSE) GO TO 20
00150 DX=(DT*V)*COS(ALAST)
00160 DY=(DT*V)*SIN(ALAST)
00170 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
00180 IF(RNOW.GT.R) GO TO 20
00190 XNOW=XLAST+DX
00200 YNOW=YLAST+DY
00210 GO TO 10
00220 20 CONTINUE
00230 IND=1
00240 TN=1.
00250 ALPHA=6.28319*RANF(XX)
00260 PHY=ALPHA+ALAST
00270 PHY=AMOD(PHY,6.28319)
00280 DX=(DT*V)*COS(PHY)
00290 DY=(DT*V)*SIN(PHY)
00300 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
00310 IF(RNOW.GT.R) GO TO 20
00320 ALAST=PHY
00330 XNOW=XLAST+DX
00340 YNOW=YLAST+DY
00350 10 TN=TN+1.
00360 RETURN
00370 END
```

STATICS 06/10/71.

```
00100  PROGRAM STATICS(INPUT,OUTPUT,TAPE4,TAPES,
00110+ TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12)
00120  13  CONTINUE
00130  PRINT,*ENTER FILE NO*
00140  READ,K
00150  GO TO (3,3,3,4,5,6,7,8,9,10,11,12),K
00160  3  CONTINUE
00170  PRINT,*ERR-FILE NOT USE ,TRY AGAIN*
00180  GO TO 13
00190  4  NAME=5HTAPE4
00200  GO TO 14
00210  5  NAME=5HTAPE5
00220  GO TO 14
00230  6  NAME=5HTAPE6
00240  GO TO 14
00250  7  NAME=5HTAPE7
00260  GO TO 14
00270  8  NAME=5HTAPE8
00280  GO TO 14
00290  9  NAME=5HTAPE9
00300  GO TO 14
00310  10 NAME=6HTAPE10
00320  GO TO 14
00330  11 NAME=6HTAPE11
00340  GO TO 14
00350  12 NAME=6HTAPE12
00360  GO TO 14
00370  14  CONTINUE
00380  CALL RETR(K,NAME)
00390  REWIND K
00400  READ(K,55)KOUNT
00410  55  FORMAT(I10)
00420  IF(KOUNT.NE.0) GO TO 20
00430  PRINT,*0 REC ON FILE K,TRY AGAIN*
00440  GO TO 13
00450  20  CONTINUE
00460  SUM=0.
00470  50  READ(K,30) I,TIME
00480  30  FORMAT(I10,E20.4)
00490  SUM=SUM+TIME
00500  IF(I.NE.KOUNT) GO TO 50
00510  ISUM=I
00520  IF(ISUM.EQ.KOUNT) GO TO 60
00530  PRINT,*ERROR-KOUNT NOT MATCH LAST REC NO*
00540  60  PRINT,*ISUM,SUM*
00550  PRINT,ISUM,SUM
00560  FEAN=SUM/FLOAT(ISUM)
00570  REWIND K
00580  SUM=0.
00590  80  READ(K,30) I,TIME
```

STATICS 06/10/71.

```
00600 X=(TIME-FEAN)**2
00610 SUM=SUM+X
00620 IF(I.NE.KOUNT) GO TO 80
00630 GO TO 70
00640 70 VAR=SUM/F DAT(I SUM)
00650 STAN=SQRT(VAR
00660 PRINT,*SAMPLE SIZE IS*
00670 PRINT,I SUM
00680 PRINT,*MEAN, STANDARD DEVIATION, VARIANCE*
00690 PRINT,FEAN, STAN, VAR
00700 RETURN
00710 END
```

UPDATE 06/10/71.

```
00100  PROGRAM  UPDATE(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,
00110+ TAPE4,TAPE5,TAPE6,TAPE7,TAPE8,TAPE9)
00120  2  CONTINUE
00130  PRINT,*THIS IS DATA FILE UPDATE*
00140  PRINT,*ENTER FILE NO*
00150  READ,IFILE
00160  PRINT,*ENTER,K AND KOUNT, K=1 OR 2 FOR FILE TAPE1 OR TAPE2*
00170  PRINT,*KOUNT=NUMBER OF RECORDS TO BE INSERTED*
00180  READ,K,KOUNT
00190  CALL  RETR(1,5HTAPE1)
00200  CALL  RETR(2,5HTAPE2)
00210  GO  TO  (1,1,1,4,5,6,7),IFILE
00220  1  PRINT,*ERROR-IFILE LESS THAN 4,TRY AGAIN*
00230  PRINT,IFILE
00240  GO  TO  2
00250  4  NAME=5HTAPE4
00260  GO  TO  88
00270  5  NAME=5HTAPE5
00280  GO  TO  88
00290  6  NAME=5HTAPE6
00300  GO  TO  88
00310  7  NAME=5HTAPE7
00320  GO  TO  88
00330  8  NAME=5HTAPE8
00340  GO  TO  88
00350  9  NAME=5HTAPE9
00360  GO  TO  88
00370  88  CALL  RETR(IFILE,NAME)
00380  REWIND IFILE
00390  REWIND 1
00400  REWIND 2
00410  REWIND 3
00420  120  CONTINUE
00430  READ(IFILE,101) ILAST
00440  PRINT,ILAST
00450  IF(ILAST.EQ.0)  GO  TO  888
00460  101  FORMAT(I10)
00470  NREC=KOUNT+ILAST
00480  WRITE(3,101)  NREC
00490  PRINT,NREC
00500  ICOUN=1
00510  130  CONTINUE
00520  READ(IFILE,100) I,TIME
00530  100  FORMAT(I10,E20.4)
00540  IF(ICOUN.EQ.ILAST)  GO  TO  110
00550  WRITE(3,100)  I,TIME
00560  ICOUN=ICOUN+1
00570  GO  TO  130
00580  110  WRITE(3,100)  ICOUN,TIME
00590  ICOUN=ILAST+1
```

UPDATE 06/10/71.

```
00600 150 REWIND K
00610 READ(K,101) I
00620 151 CONTINUE
00630 READ(K,100) I,TIME
00640 WRITE(3,100) ICOUN,TIME
00650 IF(ICOUN.EQ.NREC) GO TO 140
00660 ICOUN=ICOUN+1
00670 GO TO 151
00680 140 WRITE(3,100) ICOUN,TIME
00690 ENDFILE 3
00700 REWIND 3
00710 REWIND IFILE
00720 READ(3,101) NREC
00730 WRITE(IFILE,101) NREC
00740 240 READ(3,100) ICOUN,TIME
00750 IF(ICOUN.EQ.NREC) GO TO 230
00760 WRITE(IFILE,100) ICOUN,TIME
00770 GO TO 240
00780 230 WRITE(IFILE,100) ICOUN,TIME
00790 ENDFILE IFILE
00800 REWIND IFILE
00810 CALL REPL(IFILE,NAME)
00820 PRINT,*FILE UPDATE IS, LAST REC CREATED IS*
00830 PRINT,IFILE,ICOUN
00840 GO TO 99
00850 888 CONTINUE
00860 REWIND IFILE
00870 NREC=KOUNT
00880 WRITE(IFILE,101) NREC
00890 READ(K,725) KOUNT
00900 725 FORMAT(I10)
00910 ICOUN=1
00920 210 CONTINUE
00930 READ(K,100) I,TIME
00940 IF(I.EQ.KOUNT) GO TO 200
00950 IF(ICOUN.EQ.KOUNT) GO TO 200
00960 WRITE(IFILE,100) ICOUN,TIME
00970 ICOUN=ICOUN+1
00980 GO TO 210
00990 200 WRITE(IFILE,100) ICOUN,TIME
01000 PRINT,*FILE UPDATE IS, LAST REC CREATED*
01010 PRINT,IFILE,ICOUN
01020 ENDFILE IFILE
01030 REWIND IFILE
01040 CALL REPL(IFILE,NAME)
01050 99 CONTINUE
01060 END
```



FILECR 06/10/71.

```
00100 PROGRAM FILECR(INPUT,OUTPUT,TAPE1,TAPE2,
00110+ TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,TAPE11,TAPE12)
00120 DO 100 I=4,12
00130 GO TO (4,4,4,4,5,6,7,8,9,10,11,12),I
00140 4 NAME=5HTAPE1
00150 IFILE=1
00160 GO TO 110
00170 5 NAME=5HTAPE2
00180 IFILE=2
00190 GO TO 110
00200 6 NAME=5HTAPE6
00210 IFILE=6
00220 GO TO 110
00230 7 NAME=5HTAPE7
00240 IFILE=7
00250 GO TO 110
00260 8 NAME=5HTAPE8
00270 IFILE=8
00280 GO TO 110
00290 9 NAME=5HTAPE9
00300 IFILE=9
00310 GO TO 110
00320 10 NAME=6HTAPE10
00330 IFILE=10
00340 GO TO 110
00350 11 NAME=6HTAPE11
00360 IFILE=11
00370 GO TO 110
00380 12 NAME=6HTAPE12
00390 IFILE=12
00400 110 CONTINUE
00410 REWIND IFILE
00420 ICOUN=0
00430 II=1
00440 TIME=0.
00450 WRITE(IFILE,30) ICOUN
00460 30 FORMAT(I10)
00470 WRITE(IFILE,35) II,TIME
00480 35 FORMAT(I10,E20.4)
00490 ENDFILE IFILE
00500 REWIND IFILE
00510 100 CONTINUE
00520 END
```

Appendix C

PROGRAM LISTING FOR CONTINUOUS SEARCH -  
DECOY FIELD SUBMODEL (MOD 3)

MOD3 06/11/71.

```
00100 PROGRAM MOD3(INPUT,OUTPUT,TAPE1,TAPE2)
00110 DIMENSION TND(5),XD(5),YD(5),ANGD(5),IND(5),COURD(5)
00120 DIMENSION NOPD(5),ICHASD(5),VDC(5)
00130 DIMENSION TDCVA(500),TDCOY(500),KPD(500)
00140 DIMENSION VD(5)
00150 DIMENSION DTMON(5)
00160 DIMENSION IPAR(5)
00170 PRINT,*ENTER NO OF DECOYS*
00180 READ,NDECOY
00190 DO 2000 NI=1,NDECOY
00200 PRINT 2010,NI
00210 2010 FORMAT(1H ,*ENTER FOR DECOY NO*,13,2X,*COURD,VD*)
00220 READ,COURD(NI),VD(NI)
00230 2000 CONTINUE
00240 PRINT,*ENTER VC,VS,COURC,COURS*
00250 READ,VC,VS,COURC,COURS
00260 PRINT,*ENTER TC,TM,R1R,R2R*
00270 READ,TC,TM,R1R,R2R
00280 PRINT,*ENTER NO OF REPLICATIONS(NOT TO EXCEED 500)*
00290 READ,NREPLI
00300 NINT=500
00310 ITGT5=0
00320 IP1=1
00330 IP2=5
00340 R=200.
00350 PRINT,*C*
00360 PRINT,*CVA,DECOYS RANDOM START IN AREA*
00370 PRINT,*SUB RANDOM START ON BOUNDARY*
00380 PRINT,*C*
00390 PRINT,*COUNTER TIME ENCOUNTERED*
00400 CALL SECOND(X)
00410 Y=RANF(X)
00420 XX=0.
00430 TIME=0.
00440 HRS=0.
00450 KOUNT=0
00460 TOND=0
00470 KONC=0
00480 15 CONTINUE
00490 DO 662 IDY=1,NDECOY
00500 DTMON(IDY)=0.
00510 662 CONTINUE
00520 TIME=0.
00530 HRS=0.
00540 INC=0
00550 INS=0
00560 TNC=0.
00570 TNS=0.
00580 ICHASS=0
00590 DEG=57.+(17./60.)
```

MOD3      06/11/71.

```
00600 DO 2 I=1,NDECOY
00610 IND(I)=0
00620 TND(I)=0.
00630 ICHASD(I)=0
00640 2 CONTINUE
00650 200 CONTINUE
00660 ANGC=6.28319*RANF(XX)
00670 RADUS=R*RANF(XX)
00680 XC=RADUS*COS(ANGC)
00690 YC=RADUS*SIN(ANGC)
00700 TNC=TNC+1.
00710 ANGS=6.28319*RANF(XX)
00720 RAD=R
00730 XS=RAD*COS(ANGS)
00740 YS=RAD*SIN(ANGS)
00750 TNS=TNS+1.
00760 INC=1
00770 INS=1
00780 IF(NDECOY.EQ.0) GO TO 800
00790 DO 4 I=1,NDECOY
00800 CALL INDCOY(ANGC(I),XD(I),YD(I),XC,YC,XS,YS,
00810+ XX,R,VDC(I),VS,VC)
00820 TND(I)=TND(I)+1.
00830 IND(I)=1
00840 4 CONTINUE
00850 800 CONTINUE
00860 DT=0.5
00870 202 CONTINUE
00880 TIME=TIME+DT
00890 HRS=HRS+DT
00900 IC=0
00910 CALL LITBAL(TNC,XC,YC,ANGC,XNOW,YNOW,INC,
00920+ VC,COURC,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
00930+ XC,YC)
00940 XC=XNOW
00950 YC=YNOW
00960 IF(ICHASS.EQ.0) GO TO 300
00970 IF(STMON.NE.TIME) GO TO 305
00980 DO 310 I=1,NDECOY
00990 IF(ICHASD(I).EQ.1) GO TO 315
01000 310 CONTINUE
01010 PRINT,*ERR-FNCOUNTERED DFCOY NOT TURNED ON*
01020 PRINT 3,I
01030 3 FORMAT(I10)
01040 GO TO 00202
01050 315 XS=XD(I)
01060 YS=YD(I)
01070 TNS=1.
01080 ANGS=6.28319*RANF(XX)
01090 INS=1
```

MOD3      06/11/71.

```
01100  ICHASS=0
01110  GO TO 00320
01120  300 CONTINUE
01130  IC=88
01140  CALL LITBAL(TNS,XS,YS,ANGS,XNOW,YNOW,INS,VS,
01150+  COURS,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
01160+  XC,YC)
01170  XS=XNOW
01180  YS=YNOW
01190  GO TO 00320
01200  305 IF(TIME.LT.STMON) GO TO 320
01210  GO TO 00300
01220  320 CONTINUE
01230  IF(NDECOY.EQ.0) GO TO 370
01240  DO 330 I=1,NDECOY
01250  IF(ICHASD(I).EQ.1) GO TO 331
01260  IF(TIME.LT.DTMON(I)) GO TO 330
01270  IF(TIME.EQ.DTMON(I)) GO TO 330
01280  350 CONTINUE
01290  IC=I
01300  CALL LITBAL(TND(I),XD(I),YD(I),ANGD(I),XNOW,YNOW,
01310+  IND(I),VD(I),COURD(I),XX,VC,VS,VDC,XD,YD,
01320+  IC,NDECOY,XC,YC)
01330  XD(I)=XNOW
01340  YD(I)=YNOW
01350  GO TO 00330
01360  331 CONTINUE
01370  IF(TIME.LT.DTMON(I)) GO TO 350
01380  IF(TIME.EQ.DTMON(I)) GO TO 345
01390  IF((TIME-DTMON(I)).LT.DT) GO TO 342
01400  PRINT,*ERR-ICHASD=1,DTMON.LT.TIME*
01410  PRINT,TIME,DTMON(I),I
01420  ICHASD(I)=0
01430  GO TO 00330
01440  342 CONTINUE
01450  ICHASD(I)=0
01460  ICHASS=0
01470  DTMON(I)=DTMON(I)+TM
01480  XS=XD(I)
01490  YS=YD(I)
01500  INS=1
01510  TNS=2.
01520  ANGS=6.28319*RANF(XX)
01530  GO TO 00202
01540  345 CONTINUE
01550  ICHASD(I)=0
01560  DTMON(I)=DTMON(I)+TM
01570  ICHASS=0
01580  XS=XD(I)
01590  YS=YD(I)
```

```

01600 ANG5=6.28319*FANF(XX)
01610 INS=1
01620 TNS=2.
01630 GO TO 00330
01640 330 CONTINUE
01650 370 CONTINUE
01660 420 CONTINUE
01670 CALL MEFT(VC,VS,XC,YC,XS,YS,IFN)
01680 IF(IFN.EQ.1) GO TO 100
01690 GO TO 00390
01700 390 CONTINUE
01710 IF(ICHASS.EQ.1) GO TO 202
01720 IF(NDECOY.EQ.0) GO TO 202
01730 ISAVE=0
01740 DO 440 I=1,NDECOY
01750 IF(ICHASD(I).EQ.1) GO TO 202
01760 IF(TIME.LE.DTMON(I)) GO TO 440
01770 CALL MEFT(VDC(I),VS,XD(I),YD(I),XS,YS,IFN)
01780 IF(IFN.EQ.0) GO TO 440
01790 DISM=SQRT((XD(I)-XS)**2+(YD(I)-YS)**2)
01800 ISAVE=I
01810 440 CONTINUE
01820 IF(ISAVE.EQ.0) GO TO 202
01830 DO 450 I=1,NDECOY
01840 IF(I.EQ.ISAVE) GO TO 450
01850 IF(TIME.LE.DTMON(I)) GO TO 450
01860 CALL MEFT(VDC(I),VS,XD(I),YD(I),XS,YS,IFN)
01870 IF(IFN.EQ.0) GO TO 450
01880 DIS=SQRT((XD(I)-XS)**2+(YD(I)-YS)**2)
01890 IF(DIS.GT.DISM) GO TO 450
01900 DISM=DIS
01910 ISAVE=I
01920 450 CONTINUE
01930 DTMON(ISAVE)=TIME+TC
01940 ICHASD(ISAVE)=1
01950 STMON=TIME+TC
01960 ICHASS=1
01970 GO TO 00460
01980 100 KOUNT=KOUNT+1
01990 KONC=KONC+1
02000 TDCVA(KONC)=HRS
02010 IPRINT=IP1+1
02020 IF(IPRINT.NE.KOUNT) GO TO 1000
02030 IP1=IP1+1
02040 PRINT 95,KOUNT,TDCVA(KONC)
02050 95 FORMAT(I10,E11.4)
02060 WRITE(1,95) KOUNT,TDCVA(KONC),TIME
02070 1000 CONTINUE
02080 IF(KOUNT.EQ.NREFL1) GO TO 97
02090 ISTAT=IP2*10

```

MOD3 06/11/71.

```
02100 IF(ISTAT.NE.KOUNT) GO TO 1010
02110 CALL STAT(TDCVA,KONC,FEANC,STDEV,VARC)
02120 PRINT,*MEANC,STDEV,VARC*
02130 PRINT,FEANC,STDEV,VARC
02140 CALL FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
02150 PRINT,*FREQUENCY GT 5000.-C A*
02160 PRINT,ITGT5
02170 IF(KOND.EQ.0) GO TO 15
02180 CALL FREQ(TDCOY,KOND,KOND,KOND,NINT,ITGT5)
02190 PRINT,*FREQUENCY GT 5000.-DECOY*
02200 PRINT,ITGT5
02210 IP2=IP2+5
02220 GO TO 15
02230 460 CONTINUE
02240 KOND=KOND+1
02250 TDCOY(KOND)=HRS
02260 KPD(KOND)=KOUNT+1
02270 GO TO 02202
02280 1010 CONTINUE
02290 GO TO 15
02300 97 CONTINUE
02310 CALL STAT(TDCVA,KONC,FEANC,STDEV,VARC)
02320 PRINT,*MEANC,STDEV,VARC*
02330 PRINT,176,FEANC,STDEV,VARC
02340 176 FORMAT(3F11.4)
02350 CALL FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
02360 PRINT,*FREQUENCY GT 5000.-CVA*
02370 PRINT,ITGT5
02380 GO TO 999
02390 IF(KOND.EQ.0) GO TO 999
02400 CALL FREQ(TDCOY,KOND,KOND,KOND,NINT,ITGT5)
02410 PRINT,*FREQUENCY GT 5000.-DECOY*
02420 PRINT,ITGT5
02430 IF(KOND.EQ.0) GO TO 999
02440 CALL STAT(TDCOY,KOND,FEAND,STDEV,VAR)
02450 PRINT,*MEAND,STDEV,VAR*
02460 PRINT,176,FEAND,STDEV,VAR
02470 999 CONTINUE
02480 111 CONTINUE
02490 END
```

MOD3 06/11/71.

```
02500 SUBROUTINE STAT(TIME, ICOUT, FEAN, STDEV, VAR)
02510 DIMENSION TIME(500)
02520 SUM=0.
02530 DO 5 I=1, ICOUT
02540 SUM=SUM+TIME(I)
02550 5 CONTINUE
02560 FEAN=SUM/FLOAT(ICOUT)
02570 VAR=0.
02580 DO 10 I=1, ICOUT
02590 VAR=VAR+(TIME(I)-FEAN)**2
02600 10 CONTINUE
02610 VAR=VAR/FLOAT(ICOUT)
02620 STDEV=SQRT(VAR)
02630 RETURN
02640 END
```

ONLY AVAILABLE COPY



```

02650  SUBROUTINE  LITRAL(TN,XLAST,YLAST,ALAST,XNOW,YNOW,IND,
02660+  V,COURSE,XX,VC,VS,VDC,XD,YD,IC,NDECOY,
02670+  XC,YC)
02680  DIMENSION DYMR(5),SUMR(5)
02690  DIMENSION XD(5),YD(5),VDC(5)
02700  DT=.5
02710  R=200.
02720  IF(IC.EQ.0) GO TO 20
02730  IF(IC.EQ.88) GO TO 30
02740  IF(IC.LE.NDECOY) GO TO 25
02750  PRINT,*ERR,IC INCORRECT*
02760  PRINT,IC
02770  RETURN
02780  20  CONTINUE
02790  IF(IND.EQ.0) GO TO 15
02800  CALL XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
02810+  XNOW,YNOW,IND,ANOW,IANG)
02820  45  DO 40 I=1,NDECOY
02830  DYMR(I)=SQRT((XD(I)-XNOW)**2+(YD(I)-YNOW)**2)
02840  RCS=R2R
02850  VP=VDC(I)
02860  SUMR(I)=RCS+R1R
02870  IF(DYMR(I).LE.SUMR(I)) GO TO 35
02880  40  CONTINUE
02890  IF(IANG.EQ.0) GO TO 999
02900  ALAST=ANOW
02910  IANG=0
02920  GO TO 999
02930  35  IND=0
02940  15  CALL NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
02950+  XNOW,YNOW,IANG)
02960  TN=2.
02970  IND=1
02980  IANG=1
02990  GO TO 45
03000  25  CONTINUE
03010  IF(IND.EQ.0) GO TO 50
03020  CALL XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
03030+  XNOW,YNOW,IND,ANOW,IANG)
03040  RCS=R2R
03050  60  DYMR(IC)=SQRT((XC-XNOW)**2+(YC-YNOW)**2)
03060  VP=VDC(IC)
03070  SUMR(IC)=RCS+R1R
03080  IF(DYMR(IC).LE.SUMR(IC)) GO TO 55
03090  IF(IANG.EQ.0) GO TO 999
03100  ALAST=ANOW
03110  IANG=0
03120  GO TO 999
03130  55  CONTINUE
03140  IND=0

```

MOD3 06/11/71.

```
03150 50    CONTINUE
03160 CALL NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
03170+ XNOW,YNOW,IANG)
03180 IND=1
03190 TN=2.
03200 IANG=1
03210 GO TO 60
03220 30    CONTINUE
03230 IF(IND.EQ.0) GO TO 65
03240 CALL XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,XNOW,
03250+ YNOW,IND,ANOW,IANG)
03260 IF(IANG.EQ.0) GO TO 999
03270 ALAST=ANOW
03280 IANG=0
03290 GO TO 999
03300 65    CONTINUE
03310 CALL NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,
03320+ XNOW,YNOW,IANG)
03330 TN=2.
03340 IND=1
03350 ALAST=ANOW
03360 IANG=0
03370 999    CONTINUE
03380 RETURN
03390 END
```

MOD3 06/11/71.

```
03400 SUBROUTINE MFET(VC,VS,XC,YC,XS,YS,IEN)
03410 R0=60.
03420 RANGE=SQRT((XC-XS)**2+(YC-YS)**2)
03430 IF(RANGE.GT.R0) GO TO 10
03440 IEN=1
03450 GO TO 00020
03460 10 IEN=0
03470 20 CONTINUE
03480 RETURN
03490 END
```

MOD3      06/11/71.

```
03500  SUBROUTINE  INDCOY(ANGD1,XD1,YD1,XC,YC,XS,YS,XX,R,  
03510+  VDC,VS,VC)  
03520  5      CONTINUE  
03530  ANGD1=6.28319*RANF(XX)  
03540  RAD=R*RANF(XX)  
03550  XD1=RAD*COS(ANGD1)  
03560  YD1=RAD*SIN(ANGD1)  
03570  CALL  MEET(VDC,VS,XD1,YD1,XS,YS,IEN)  
03580  IF(IEN.EQ.1)  GO  TO  5  
03590  RDS=R1R  
03600  RCS=R2R  
03610  SUMR=RDS+RCS  
03620  RCVADY=SQRT((XC-XD1)**2+(YC-YD1)**2)  
03630  IF(RCVADY.EQ.0)  RETURN  
03640  GO  TO  5  
03650  RETURN  
03660  END
```

MOD3 06/11/71.

```
03670 SUBROUTINE XYNOW(TN,DT,V,XLAST,YLAST,COURSE,ALAST,R,
03680+ XNOW,YNOW,IND,ANOW,IANG)
03690 5 CONTINUE
03700 FLEN=TN*DT*V
03710 IF(FLEN.GT.COURSE) GO TO 10
03720 DX=(DT*V)*COS(ALAST)
03730 DY=(DT*V)*SIN(ALAST)
03740 RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
03750 IF(RNOW.GT.R) GO TO 10
03760 XNOW=XLAST+DX
03770 YNOW=YLAST+DY
03780 TN=TN+1.
03790 IANG=0
03800 GO TO 99
03810 10 CONTINUE
03820 CALL NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,ANOW,XNOW,YNOW,IANG)
03830 TN=2.
03840 IND=1
03850 IANG=1
03860 99 CONTINUE
03870 RETURN
03880 END
```

ORIGINAL COPY

MOD3      06/11/71.

```
03890  SUBROUTINE  NPHY(XX,R,V,DT,XLAST,YLAST,ALAST,
03900+  ANOV,XNOW,YNOW,IANG)
03910  20      CONTINUE
03920  ALPHA=6.28319*RANF(XX)
03930  PHY=ALPHA+ALAST
03940  PHY=AMOD(PHY,6.28319)
03950  DX=(DT*V)*COS(PHY)
03960  DY=(DT*V)*SIN(PHY)
03970  RNOW=SQRT((DX+XLAST)**2+(DY+YLAST)**2)
03980  IF(RNOW.GT.R)  GO  TO  20
03990  ANOV=PHY
04000  XNOW=XLAST+DX
04010  YNOW=YLAST+DY
04020  RETURN
04030  END
```

MOD3      06/11/71.

```
04040  SUBROUTINE  FREQ(TDCVA,KONC,ISTAT,NREPLI,NINT,ITGT5)
04050  DIMENSION  TDCVA(500),IFRE(500),TCOM(500)
04060  DIMENSION  IFREC(500)
04070  ITGT5=0
04080  IF(ISTAT.LT.NREPLI)  GO  TO  5
04090  ISUM=NREPLI
04100  GO  TO  6
04110  5  ISUM=ISTAT
04120  6  TINT=10.
04130  TMAX=0.
04140  DO  7  J=1,NINT
04150  IFRE(J)=0
04160  IFREC(J)=0
04170  7  CONTINUE
04180  DO  10  I=1,ISUM
04190  DO  20  J=NINT
04200  TCOMPA=FLOAT(I)*TINT
04210  IF(TDCVA(I).GT.5000.)  GO  TO  40
04220  IF(TDCVA(I).GT.TCOMPA)  GO  TO  20
04230  IFRE(J)=IFRE(J)+1
04240  IF(TCOMPA.LT.TMAX)  GO  TO  10
04250  TMAX=TCOMPA
04260  GO  TO  10
04270  40  ITGT5=ITGT5+1
04280  GO  TO  10
04290  20  CONTINUE
04300  10  CONTINUE
04310  IP=TMAX/TINT+1.
04320  DO  30  I=1,IP
04330  TCOM(I)=FLOAT(I)*TINT
04340  30  CONTINUE
04350  IFREC(I)=IFRE(I)
04360  DO  110  I=2,IP
04370  IFREC(I)=IFRE(I)+IFREC(I-1)
04380  110  CONTINUE
04390  PRINT,*,TIME INTERVAL,FREQUENCY,CUM F*
04400  DO  100  I=1,IP
04410  IF(IFRE(I).EQ.0)  GO  TO  100
04420  PRINT 35,TCOM(I),IFRE(I),IFREC(I)
04430  35  FORMAT(1H,2X,E11.2,2X,2I10)
04440  100  CONTINUE
04450  RETURN
04460  END
```

# DISTRIBUTION LIST

	<u>No. of Copies</u>
Director of Defense Research & Engineering Department of Defense Washington, D. C. 20301 (Tactical Systems Plans and Analysis)	1
Advanced Research Projects Agency Department of Defense Washington, D.C. 20301 (Technical Library)	1
Chief of Naval Operations Department of the Navy Washington, D.C. 20350 (Op-952)	1
(Op-981)	1
(Op-96, SEAMIX Project Officer)	1
Office of Naval Research Department of the Navy Arlington, Virginia 22217 (Code 462)	2
(Code 434)	1
Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101	1
Headquarters, Marine Corps Washington, D.C. 20380 (Dr. A.L. Slafkosky, Science Advisor)	1
Naval Air Systems Command Department of the Navy Washington, D.C. 20360 (AIR-370)	1
(AIR- 3362)	1



No. of Copies

Naval Ship Systems Command Department of the Navy Washington, D.C. 20360 (SHIP-902)	1
Commander Antisubmarine Warfare Force U.S. Pacific Fleet FPO San Francisco, California 96610 (CDR Fishburn)	1
Naval Weapons Laboratory (Code KCU) Dahlgren, Virginia 22448	1
Naval Ordnance Laboratory Silver Spring, Maryland 20910 (Technical Library)	1
Naval Academy Annapolis, Maryland 21402 (Technical Library)	1
Naval Postgraduate School Monterey, California 93940 (Department of Operations Research and Administrative Sciences)	2
Naval War College Newport, Rhode Island 02844 (Technical Library)	1
Naval Research Laboratory Washington, D.C. 20390 (Code 2620) (Code 2629/ONR-L)	2 2
Naval Air Development Center Warminster, Pennsylvania 18974 (Mr. Madera)	1
Naval Ship Research and Development Center (Code OH74) Washington, D.C. 20034	1

	<u>No. of Copies</u>
Naval Undersea Research and Development Center San Diego, California 92152	1
New London Laboratory Naval Underwater Systems Center New London, Connecticut 06320 (Mr. C. King)	1
Applied Physics Laboratory Johns Hopkins University 8621 Georgia Avenue Silver Spring, Maryland 20910	1
Center for Naval Analysis 1401 Wilson Boulevard Arlington, Virginia 22209 (Library)	1
(Dr. S. Shear)	1
Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Arthur D. Little, Inc. Acorn Park Cambridge, Massachusetts 02139 (Dr. J. M. Dobbie)	1
Ordnance Research Laboratory State College, Pennsylvania 16801 (Dr. Janetta)	1
The Rand Corporation 1700 Main Street Santa Monica, California 90401	1
Santa Barbara Analysis and Planning Corp. P.O. Box 3798 Santa Barbara, California 93105	1
Ultrasystems, Inc. 500 Newport Center Drive Newport Beach, California 92660	1

No. of Copies

Daniel H. Wagner Associates, Inc.  
Station Square One  
Paoli, Pennsylvania 19301

1

Presearch Incorporated  
8720 Georgia Avenue  
Silver Spring, Maryland 20910

1